Claims

I claim:

1. A method for analyzing a scalability of an application server, comprising:

providing a test configuration, wherein the test configuration includes a component quantity, a node quantity, a server quantity and a resource quantity;

creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity;

deploying, installing and starting the application on the application server;

creating a client program based on the test configuration;

running the client program against the application; and

monitoring performance metrics during the deploying, installing, starting and running steps to verify the scalability of the application server.

2. The method of claim 1, further comprising generating output based on the performance metrics.

3. The method of claim 2, wherein the output illustrates the performance metrics for the provided component quantity versus a potential performance for the application with a different quantity of components.

4. The method of claim 1, wherein the monitoring step comprises:

monitoring deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage and an application deploy time during the deploying step;

monitoring installation metrics selected from the group consisting of a memory consumption, a processor usage and an application install time during the installing step;

monitoring start metrics selected from the group consisting of a memory increase, a processor usage and an application start time during the starting step; and

monitoring client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time and an application run time during the running step.

5. The method of claim 1, wherein the application server is implemented on a node that is part of a hierarchy of nodes that match the provided node quantity.

6. The method of claim 1, wherein the client program simulates use of the application by a quantity of clients that match the provided component count.

7. The method of claim 1, wherein the step of creating the application comprises using a component template to generate the quantity of components.

8. The method of claim 1, wherein the application is deployed using a deploy tool of the application server.

9. The method of claim 1, wherein the step of running the client program comprises conducting a database transaction.

10. A system for analyzing a scalability of an application server, comprising:

a test configuration system for providing a test configuration, wherein the test configuration includes a component quantity, a node quantity, a server quantity and a resource quantity;

an application generation system for creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity, and wherein the application is deployed, installed and started on the application server;

a client program system for creating a client program based on the test configuration, and for running the client program against the application; and

a metric monitoring system for monitoring performance metrics while the application is deployed, installed and started and when the client program is run to verify the scalability of the application server.

11. The system of claim 10, further comprising an output system for generating output based on the performance metrics.

12. The system of claim 11, wherein the output illustrates the performance metrics for the provided component quantity versus a potential performance for the application with a different quantity of components.

13. The system of claim 10, wherein the metric monitoring system monitors:

deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage and an application deploy time as the application is being deployed;

installation metrics selected from the group consisting of a memory consumption, a processor usage and an application install time as the application is being installed;

start metrics selected from the group consisting of a memory increase, a processor usage and an application start time as the application is being started; and

client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time and an application run time as the client program runs.

14. The system of claim 10, wherein the application server is implemented on a node that is part of a hierarchy of nodes that match the provided node quantity.

15. The system of claim 10, wherein the client program simulates use of the application by a quantity of clients that match the provided component count.

16. The system of claim 10, wherein the quantity of components is generated using a component template.

17. The system of claim 10, wherein the application is deployed using a deploy tool of the application server.

18. The system of claim 10, wherein the client program conducts a database transaction upon being run.

19. The system of claim 10, wherein the test configuration system provides a graphical user interface for inputting the test configuration.

20. A program product stored on a recordable medium for analyzing a scalability of an application server, which when executed, comprises:

program code for providing a test configuration, wherein the test configuration includes a component quantity, a node quantity, a server quantity and a resource quantity;

program code for creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity, and wherein the application is deployed, installed and started on the application server;

program code for creating a client program based on the test configuration, and for running the client program against the application; and

program code for monitoring performance metrics while the application is deployed, installed and started and when the client program is run to verify the scalability of the application server.

21. The program product of claim 20, further comprising program code for generating output based on the performance metrics.

22. The program product of claim 21, wherein the output illustrates the performance metrics for the provided component quantity versus a potential performance for the application with a different quantity of components.

23. The program product of claim 20, wherein the program code for monitoring monitors:

deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage and an application deploy time as the application is being deployed;

installation metrics selected from the group consisting of a memory consumption, a processor usage and an application install time as the application is being installed;

start metrics selected from the group consisting of a memory increase, a processor usage and an application start time as the application is being started; and

client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time and an application run time as the client program runs.

24. The program product of claim 20, wherein the application server is implemented on a node that is part of a hierarchy of nodes that match the provided node quantity.

25. The program product of claim 20, wherein the client program simulates use of the application by a quantity of clients that match the provided component count.

26. The program product of claim 20, wherein the quantity of components is generated using a component template.

27. The program product of claim 20, wherein the application is deployed using a deploy tool of the application server.

28. The program product of claim 20, wherein the client program conducts a database transaction upon being run.

29. The program product of claim 20, wherein the program code for providing the test configuration provides a graphical user interface for inputting the test configuration.